Listing of claims:

1. (Previously presented) A processor comprising:
a register file;
an execution unit;
a register file cache coupled to the register file and to the execution unit; and
a write-back mechanism to move data from the register file cache to the register file.

2. (Original) The processor of claim 1, wherein the register file cache comprises a write-back portion to receive a result of an instruction executed by the execution unit.

3. (Original) The processor of claim 1, wherein the register file cache comprises a fill portion to receive an operand read from the register file.

4. (Previously presented) An apparatus comprising:
a first data storage structure to hold instruction operands;
a second data storage structure to hold instruction operands, coupled to the first data storage structure;
a logic device coupled to the first data storage structure and to the second data storage structure, to execute instructions using operands read from either the first data structure or from the second data structure; and
a write-back mechanism to move data from the first data storage structure to the second data storage structure.

5. (Original) The apparatus of claim 4, further comprising:
a data-management mechanism to move data corresponding to an operand from the second data storage structure to the logic device when the data is not present in the first data storage structure.

6. (Canceled)

7. (Previously presented) The apparatus of claim 4, wherein the write-back mechanism moves the data based on a frequency of access to the data.

8. (Original) The apparatus of claim 4, wherein the first data storage structure includes a write-back portion to which to write results of instructions executed by the logic device.

9. (Original) The apparatus of claim 5, wherein the first data storage structure includes a fill portion, and the data-management mechanism is to copy the data from the second data storage structure to the fill portion.

10. (Original) The apparatus of claim 4, wherein the first data storage structure is more ported than is the second data storage structure.

11. (Original) The apparatus of claim 4, further comprising an allocation mechanism to allocate a register in the first data structure to which to write an instruction result, wherein the allocate mechanism is to allocate the register such that the result will be written to the register only when all outstanding reads of contents of the register have completed.

12. (Original) The apparatus of claim 11, further comprising a write-back mechanism to move data from the first data storage structure to the second data storage structure, wherein the write-back mechanism is to cooperate with the allocation mechanism such that previous contents of the register will have been moved to the second data structure before the contents are overwritten by the result.

13. (Original) The apparatus of claim 4, wherein the first data storage structure comprises a first section and a second section, each of the first and second sections being divided into a plurality of subsections, wherein a subsection of the first section and a subsection of the second section have an exclusive set of write paths thereto.

14. (Original) The apparatus of claim 4, wherein the first data storage structure includes shared tracks.

15. (Previously presented) A method comprising:

arranging a register file cache to communicate with an execution unit and a register file;

searching the register file cache for an instruction operand of an instruction to be executed by the execution unit;

if the operand is found in the register file cache, reading the operand from the register file cache; and

periodically writing data from the register file cache to the register file.

16. (Original) The method of claim 15, further comprising:

if the operand is not found in the register file cache, reading the operand from the register file.

17. (Original) The method of claim 16, further comprising:

copying the operand that is read from the register file to the register file cache.

18. (Original) The method of claim 16, further comprising:

executing the instruction; and

writing a result of the instruction to the register file cache.

19. (Canceled)

20. (Original) The method of claim 19, wherein the data are written based on a least-recently-used policy.

21. (Original) The method of claim 18, further comprising:

allocating a register in the register file cache to which to write the instruction result, such that the result will be written to the register only when all outstanding reads of contents of the register have completed.

22. (Original) The method of claim 18, further comprising

allocating a register in the register file cache to which to write the instruction result;

periodically writing data from the register file cache to the register file; and

timing the allocating and the periodic writing such that previous contents of the register will have been moved to the register file before the contents are overwritten by the result.

23. (Previously presented) A system comprising:

a memory to hold instructions for execution;

a processor coupled to the memory to execute the instructions, the processor including:

a register file;

an execution unit; ~~and~~

a register file cache coupled to the register file and to the execution unit; and

a write-back mechanism to move data from the register file cache to the register file.

24. (Original) The system of claim 23, wherein the register file cache comprises a write-back portion to receive a result of an instruction executed by the execution unit.

25. (Original) The system of claim 23, wherein the register file cache comprises a fill portion to receive an operand read from the register file.